REMARKS

Claims 1-62 have been canceled, and new claims 63-77 presented.

Claims 38, 40, 41, 50, 52, 54, 55, 57, 59, 60 and 62 were rejected under 35 USC 103(a) based on US 6,785,827 to Layton et al. and US 7,080,378 to Noland et al. Claims 39, 51, 53, 56, 58 and 61 were rejected under 35 USC 103(a) based on Layton et al., Noland et al. and Bruck et al. Applicants respectfully traverse this rejection as applied to the new claims 63-77, based on the following.

In new claim 63, the real management server, based on the performance data for the first real application server **and the real data-source server**, automatically determines that the first real application server is functional but has reached a predetermined upper level of utilization. The performance data for the real data-source server indicates an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool. Also in new claim 63, the real management server selects the real data-source server to provide application data for the additional real application server and sends connection settings for the real data-source server to the additional real application server. Neither Layton et al. nor Noland et al. teach or suggest this combination of features of claim 63.

Layton et al. disclose management and control facilities for fail-over of a server (and a system for determining power supply requirements of servers by sampling power usage values at a rate based upon the criticality of the servers' availability):

> "A user that wishes to access information on information network 110 typically has a client workstation 112 that executes application programs such as a browser and an electronic mail program. Workstation 112 establishes communication link with servers 118, 120, 122 through network interface 114." Layton Column 2 lines 55-60.

"Management and control facilities 124 balance the load on servers 118, 120, 122." Layton Column 3 lines 34-35.

"Management and control facilities 124 provide fail-over capabilities. For example, if server 122 fails, the workload is distributed to one or more of the remaining servers 118, 120, 122, thereby avoiding interruption of services." Layton Column 3 lines 19-22.

"If the combined capacity for servers 118, 120, 122 is insufficient for project traffic, one or more additional servers may be added. Management and control facilities 124 also include features to redistribute tasks in the event one or more servers 118, 120, 122 go off-line due to failure or scheduled maintenance, to install and update databases, and to protect information on servers 118, 120, 122 from unauthorized access." Layton Column 3 lines 26-34.

"When a power supply fails in server 118, 120, or 122, a warning is issued to alert an operator or the user that server power is in an non-redundant state." Layton Column 3 lines 52-53.

"Referring to FIG. 2, a flowchart of a method for determining power requirements and issuing an alert when power supplies may be added or removed to meeting power redundancy requirements." Layton Column 3 lines 59-62

However, Layton et al. do not disclose or even suggest that a real management server determines that a first real application server is functional but has reached a predetermined upper level of utilization, based on the performance data for the first real application server **and performance data for the real data-source server**. (This determines whether slow performance of the first real application server is due to CPU constraint of the first real application server or delays in the real data-source server furnishing data to the first real application server needed to comply with the client request to the real application server.) Also, **Layton et al. also do not disclose or suggest that a real management server selects a real data-source server for a real application server added to the pool, nor that the real management server sends connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server**. Layton et al. fail to disclose these key features of control by a management server. The Examiner stated, "Layton et al. fails to teach that the real management server automatically sending connection settings for the real data-source server to the additional real application server to configure the additional real application server to send subsequent requests for application data to the real data-source server." Noland et al. do not fill both of these gaps. Noland et al. disclose:

> Noland et al. disclose:
>
> "If some predefined resource in the virtual servers, such as CPU usage or a request queue into the cluster of servers, should reach a critical threshold, then one or more new virtual server(s) are automatically deployed. These new virtual servers are identified as additional virtual servers and can be constructed to perform identical services as the other servers in the cluster." Noland Column 1 line 65 to Column 2 line 4.

"If the deployed virtual servers are close to near-term service saturation, the software will create and deploy a new virtual server with identical applications (step 504), and then direct the service request to this new server (step 505). This new virtual server can be deployed and activated via the method depicted in FIG. 3. More than one new virtual server may be deployed and activated based on the saturation condition that is detected (e.g., rate of saturation, number of servers approaching saturation simultaneously, etc.)." Noland et al. Column 5 lines 24-36.

"Referring to FIG. 3, a flowchart illustrating the process of deploying a virtual server by means of an automated software solution is depicted in accordance with the present invention. The process begins when the user logs into the software (step 301) and selects definition criteria for the newly deployed system image (step 302). The definition criteria include the following: a pool of TCP/IP addresses that the software assigns to newly deployed virtual servers, a pool of names the software assigns to the new virtual servers, and a model image that is used as a target image for the creation and deployment of the new virtual server. The model image is the current contents of memory, including the operating system and running programs. Every new virtual server is an exact copy of the model image, except for the dynamic network and server definitions that identify the new server as a unique entity. The user verifies the definition criteria and clicks a "submit" link (step 303). This link contained an imbedded common sequence that requests the software to rapidly deploy a new virtual server. Alternatively, the step of deploying the new virtual server may be automated and triggered by specified event, e.g. reaching a saturation threshold on currently deployed servers (as described in more detail below). Furthermore, the new virtual server can actually be a subset of the original server that focuses on a critical portion of the process. This can be done in several ways. For example, the virtual server can be composed of a series of linked server processes that are individually utilized in the overall server process. When one of the sub processes becomes a bottleneck, that sub process could be cloned as necessary to eliminate the bottleneck. In response to the request from the user, the software automatically updates the VM system directory to include the new virtual server (step 304), prepares the virtual

server media (disk allocations) (step 305), propagates the server model image into the new virtual server (step 306), updates the new image with local identification parameters (step 307), and then boots the new server (step 308). After the new virtual server is deployed, the end user simply clicks another link in order to interface with the new server (step 309). Alternatively, the new server may be automatically integrated into a preexisting server cluster. The entire process of deploying a new virtual server by means of the present invention can be fully automated. When done manually, it takes less than five minutes." Noland et al. Column 3l line 56 to Column 4 line 33.

"Referring now to FIG. 4, a schematic diagram illustrating the architecture of the virtual machine environment of the automated software solution is depicted in accordance with the present invention. The present example is described within the context of the SnapVantage software solution, but it should be pointed out that the features of the present invention may be implemented by means of other software solutions. SnapVantage VM server 401 is a Virtual service machine that manages the cloning process of Linux images, i.e. Model Images 405 and 406. This cloning process uses a Shared Virtual Array Administrator (SVA) 407 in order to create array of cloned virtual servers 408. SnapVantage runs disconnected and communicates to clients 409 and 410 via TCP/IP 404. The SnapVantage web server 402 is the location of the web pages used by the SnapVantage GUI on client 409, and executes under a local Apache (or other) web server. The local deployment application 403 is the user created code imbedded in local web pages that drives specific SnapVantage functions. This component is deployed in environments that choose to allow end users to define a new virtual server. Referring now to FIG. 5, a flowchart illustrating the process of load balancing among virtual servers is depicted in accordance with the present invention. The present invention provides a software-based solution that utilizes a communication mechanism in either direction and monitors the load status of deployed servers." Noland et al. Column 4 line 36 to Column 5 line 4.

Like Layton et al., Noland et al. do not disclose or even suggest that a real management server determines that a first real application server is functional but has reached a predetermined upper level of utilization, based on the performance data for the first real application server **and performance data for the real data-source server**, where the performance for the real data-source server indicates an amount of utilization of the real data-source server in providing application data to one or more of the real application servers in the pool. Therefore, the rejection under 35 USC 103(a) based on Layton et al. and Noland et al. should be withdrawn.

Also, Noland et al. pertain to a virtual machine environment, where virtual servers are on the same real computer are formed by cloning. So, Noland et al do not automatically add another real server to a cluster of real servers, as recited in claim 63. There would be no motivation to combine Layton et al. with Noland et al. because of the differences between a virtual machine environment and a real machine environment. This is further reason to withdraw the rejection under 35 USC 103 based on Layton et al. and Noland et al.

Bruck et al. do not fill the foregoing gap of Layton et al. and Noland et al. Bruck et al. disclose:

> "A distributed server cluster for computer network data traffic dynamically reconfigures traffic assignments among multiple server machines for increased network availability. **If one of the servers becomes unavailable**, traffic assignments are moved among the multiple servers such that network availability is substantially unchanged. The front-layer servers of the server cluster communicate with each other such that automatic, dynamic traffic assignment reconfiguration occurs in response to machines being added and deleted from the cluster, with no loss in functionality for the cluster overall, in a process that is transparent to network users, thereby providing a distributed server system functionality that is scalable. Thus, operation of the distributed server cluster remains consistent as machines are added and deleted from the cluster. Each machine of the distributed cluster can continue with any applications that may be running, such as for implemented its server functioning, while participating in the distributed server cluster

and dynamic reconfiguration processing of the present invention. In this way, the invention substantially maintains network availability regardless of **machine failures**, so that there is a single point of failure and no lapse in server cluster functionality." (Emphasis added) Bruck et al. Column 3 lines 19-40.

"The operation of the servers on both layers is monitored, and when a server **failure** at either layer is detected, the system automatically shifts network traffic from the failed machine to one or more operational machines, reconfiguring front layer servers as needed without interrupting operation of the server system. The server system automatically accommodates additional machines in the server cluster, without service interruption." (Emphasis added) Column 2 lines 47-54.

Thus, Bruck et al. are concerned with responding to machine **failures** by re-routing network traffic to the operational machines in the cluster. Bruck et al. do not disclose or even suggest that a real management server determines that a first real application server is functional but has reached a predetermined upper level of utilization, based on the performance data for the first real application server **and performance data for the real data-source server**.

US 5,951,694 to Choquier et al., cited by the Examiner in the prior Office Action, does not fill the foregoing gap of Layton et al., Noland et al. and Bruck et al. Choquier et al. do not disclose or even suggest that a real management server determines that a first real application server is functional but has reached a predetermined upper level of utilization, based on the performance data for the first real application server **and performance data for the real data-source server**. Rather Choquier et al. add an application server to the cluster based on CPU utilization of the existing application servers in the cluster (not based on performance of a real data-source server):

"The CPU LOAD and a CPU INDEX for a server 120 indicate the available processing power for the server, and are thus useful for identifying the servers 120 that are most capable of handling new service sessions." Choquier et al. Column 11 lines 36-38.

"Loads placed on particular services (or equivalently, particular service groups) may fluctuate relative to one another on daily basis duke to fluctuations in usage of different services. Usage of a Stock Quote service, for example, ma peak during regular business hours, while usage of a CHAT service may peak at night or on weekends. To accommodate for such fluctuations in service usage levels, the on-line services network 100 allocates servers 120 to service groups based on service loads." Choquier et al. Column 23 lines 26-35.

"FIG. 13 is a high level flow chart of a preferred routine for monitoring the load on a particular service, service i, and for determining whether a server 120 should be added to or extracted from the corresponding service group, service group i. The routine is preferably run continuously (on a service-by-service basis) on a dedicated administrative server 134 that receives the periodically broadcasted service map 136. With reference to blocks 1300 and 1302, the service map 136 is initially accessed to identify the servers 120 currently allocated to the service group, and to read the CPU LOAD values of each identified server. These CPU LOAD values are then averaged to obtain the load of the service group, SRVGRP_LOAD. With reference to block 1304, the SRVGRP_LOAD values calculated over the last N service maps are then averaged to obtain the average load of the service group, AVG_SRVGRP_LOAD, over a predetermined period of time (defined by N). The effect of this step is to filter out short-term fluctuations in the load of the service group, so that servers 120 will not unnecessarily be taken from or added to the service group. The variable N can be set to different values for different services." Choquier et al. Column 24 lines 13-33.

Thus, Choquier et al. do not disclose or even suggest that a real management server determines that a first real application server is functional but has reached a predetermined upper level of utilization, based on the performance data for the first real application server **and performance data for the real data-source server**. Rather Choquier et al. add an application server to the cluster based on CPU utilization of the existing application servers in the cluster (not based on performance of a real data-source server). Therefore, Choquier et al. do not fill the foregoing gap of Layton et al., Noland et al. and Bruck et al.

Claim 64 depends on claim 63 and further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al. by reciting that in response to the first real application server reaching a predetermined upper level of utilization, the real management server automatically sends to the additional real application server port settings for the real data-source server to communicate with the real data-source server to obtain application data from the real data-source server. Neither Layton et al., Noland et al., Bruck et al. nor Choquier et al. teaches or suggests this feature of claim 64.

Claim 67 depends on claim 63 and further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al. by reciting that in response to the first real application server reaching a predetermined upper level of utilization, the real management server automatically sends to the additional real application server a description of an installation path for the real data-source server to support communication with the additional real application server. Neither Layton et al., Noland et al., Bruck et al. nor Choquier et al. teaches or suggests this feature of claim 67.

Independent claim 68 distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al. for the same reasons that claim 63 distinguishes thereover.

Claims 69-72 depend on claim 68 and therefore, distinguish over Layton et al., Noland et al., Bruck et al. and Choquier et al. for the same reasons that claim 68 distinguishes thereover.

Claim 69 further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al. for the same reasons that claim 64 further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al.

Claim 72 further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al. for the same reasons that claim 67 further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al.

Independent claim 73 distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al. for the same reasons that claim 63 distinguishes thereover.

Claims 74-77 depend on claim 73 and therefore, distinguish over Layton et al., Noland et al., Bruck et al. and Choquier et al. for the same reasons that claim 73 distinguishes thereover.

Claim 74 further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al. for the same reasons that claim 69 further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al.

Claim 77 further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al. for the same reasons that claim 72 further distinguishes over Layton et al., Noland et al., Bruck et al. and Choquier et al.

Based on the foregoing, Applicants request allowance of the present patent application as amended above.

Respectfully submitted,

Dated: July 22, 2009
Phone: 607-429-4368
Fax: 607-429-4119

/Arthur J. Samodovitz/
Arthur J. Samodovitz
Reg. No. 31,297